AD-A104 889 BATTELLE COLUMBUS LABS OH F/G 9/2

ENHANCEMENTS AND APPLICATIONS OF AN INTERACTIVE COMPUTER GRAPHI--ETC(U)

SEP 79 R E ROSENTHAL DAAG29-75-D-0100

UNCLASSIFIED NL

1 OF 1

AD A104889

END

DATE FILMED

10 81

DTIC

(15)

AD A104889

ENHANCEMENTS AND APPLICATIONS OF AN INTERACTIVE COMPUTER GRAPHICAL LOCATION ANALYSIS SYSTEM

Richard E. Rosenthal

Associate Professor of Management Science
The University of Tennessee
Knoxville, Tennessee 37916

DTIC
ELECTE
OCT 1 1981
S D
H

submitted to:

THE ARMY INSTITUTE FOR RESEARCH IN MANAGEMENT
INFORMATION AND COMPUTER SCIENCE

and

BATTELLE COLUMBUS LABORATORIES

under

Scientific Services Agreement, Contract DAAG29-75-D-0100

Delivery Order No. 1278

The views, opinions and findings contained in this report are those of the author and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

September, 1979

DTIC FILE COPY

Abstract

This is a report of research conducted in the Summer of 1979 on enhancements and applications of an interactive computer graphical location analysis system. The work was performed under a STAS agreement for AIRMICS, and it grew out of earlier work by the author and Stephen D. Brady. The STAS contract specified three specific tasks which were executed as follows:

a. The Brady/Rosenthal interactive location algorithm was transferred from an elementary implementation (Tektronix 4010 graphics terminal with DEC 10 host computer) to an advanced implementation on a Chromatics CG1999 color graphics microcomputer. Speed, accuracy, user convenience and program applicability were substantially enhanced as a result of this conversion.

b. Potential applications of the interactive location analysis program as an Army decision support system were identified as a result of a system demonstration with members of the Computer Systems Command Support Group. There were also identified five needs for improvement in the system: (i) ability to solve multifacility problems, (ii) general weighting functions, (iii) network modeling, (iv) general objective functions, and (v) high-specificity pre/postprocessors. The first two of these needs have been met with algorithmic enhancements in the new implementation.

c. Some suggestions have been given for meeting the remaining three needs. In addition, a field experiment for demonstrating, evaluating and further improving the system has been proposed.

CONTENTS

Accession For

SO on file Form

By

Distribution/

Availability

A

ENHANCEMENTS AND APPLICATIONS OF AN INTERACTIVE COMPUTER GRAPHICAL LOCATION ANALYSIS SYSTEM

Richard E. Rosenthal

1. Background

This report, delivered to AIRMICS and Battelle Columbus Laboratories, is in fulfillment of a scientific services agreement for short term analysis service (STAS) performed by the author in the Summer of 1979. The service was requested because some members of AIRMICS staff had read a research paper entitled "Interactive Computer Graphical Solutions of Constrained Minimax Location Problems" by Stephen D. Brady (The University of Tennessee doctoral student) and the author. The AIRMICS staff members thought that this paper contained some potentially useful ideas for decision support system users in the Army. However, due to the limitations of previously available hardware, the original implementation of these ideas was too elementary to be readily transferred to the Army's working environment. Consequently, a project with the combined participation of Mr. Brady (on a laboratory research cooperative program), AIRMICS and the author was conceived to produce an enhanced implementation. The first phase of this project has been successfully completed.

1.1 Interactive Computer Graphical Location Analysis. Location analysis is the study and development of methods for determining the locations of new facilities such that the people who use the facilities derive the greatest possible benefit from them. It is a subject that dates back at least to the ancient Greeks and has captured both the theoretical and applied interest of

some of the great mathematicians of history. Several branches of mathematics have been called upon throughout the centuries to provide solutions to location problems, the most important in recent times being linear, nonlinear and combinatorial optimization [11]. (Reference numbers refer to the bibliography given in the Appendix.)

There is a great vastness and variety to the kinds of problems that have been solved in location analysis. This is due to the fact that different situations demand different definitions of what is meant by an "optimal location." Furthermore, the mathematical context of the problems change as the domain of allowable new-facility locations takes on different form. Nevertheless, the unifying theme of the great number of papers published in the last ten years in location analysis is that some form of linear, nonlinear or combinatorial optimization is the appropriate means of attack on any location problem.

The original research performed by Mr. Brady and the author, that prompted AIRMIC's interest in the work described here, was a sharp break with the pattern of developments in location analysis as described above. The facility location problem considered in that work had a realistic context, and yet the application of mathematical optimization techniques did not offer a solution. An efficient and exact solution was found with interactive computer graphics.

Interactive graphics had earlier been used productively in decision support systems, in computer-aided design and, to a limited extent, in the solution of optimization problems [13, 17-19, 21-23, 25]. However, no other reported instance of interactive graphical optimization contains a valid proof of optimality.

1.2 The Brady/Rosenthal Algorithm. The location analysis problem attacked by Mr. Brady and the author is typified by the following case. Suppose a set of radio transmitters with unequal signal intensities and with different fixed locations is to be monitored. The receiver of these signals has yet to be constructed. The problem is to choose the size and site of the receiver with two considerations entering into the decision: first, the receiver should be as small (i.e., insensitive) as possible yet still able to monitor every signal; and second, the receiver's location must not fall within a previously specified (and arbitrarily configured) forbidden zone. The rationale for the second consideration is obvious; the reasons for the first are cost, speed of construction and, possibly, vulnerability.

Let $A_1, \ldots, A_n$ be the transmitter locations (i.e., two-dimensional vectors of coordinates) and let $I_1, \ldots, I_n$ be the respective signal intensities. By the inverse square law of electromagnetic radiation, the strength at any point $x \varepsilon R^2$ of the signal originating at A_i is

$$I_i/[d(x,A_i)]^2, \tag{1}$$

where d is Euclidean distance. Thus, if the receiver is located at x, the weakest signal it must perceive has intensity equal to the

$$\underset{i=1,\ldots,n}{\text{minimum}}\ I_i/[d(x,A_i)]^2. \tag{2}$$

The problem of constructing the least sensitive receiver that is capable of detecting every signal is then to select x so as to

$$\text{maximize minimum}\ I_i/[d(x,A_i)]^2 \tag{3}$$

subject to

$$x \in X \tag{4}$$

where X is the set of feasible locations. For example, X may be the set of points of sufficient altitude that are not within hostile territory.

By a simple transformation, the "maximin" objective function (3) can be expressed equivalently as a "minimax" problem:

$$\text{minimize} \max_{i=1,\ldots,n} w_i d(x,A_i) \tag{5}$$

subject to (4), where $w_i = 1/I_i$. This formulation is in the standard form for the algorithm. The term w_i is referred to as a "weight."

Before proceeding with the algorithm, it should be noted that the model comprised of (5) and (4) adequately represents the receiver-siting problem, but it is not the most general model that the algorithm can handle. Possible generalizations are: (i) the use of rectilinear metrics instead of Euclidean for some or all of the points A_i, (ii) the use of addend terms [Appendix I, p. 11] in the objective function, and (iii) a requirement that new-facility locations be compact sets of given area rather than single points. These generalizations were reported in the work done prior to the AIRMICS contract. Section 2.3 contains other generalizations that are more recently developed.

The theory of the algorithm is the following characterization of the optimal objective function value r* and the optimal new-facility location x*.

Theorem: For r>0 define

$$C_i(r) = \{x : w_i d(x,A_i) \leq r\} \tag{6}$$

and

$$C(r) = \bigcap_{i=1}^{n} C_i(r). \tag{7}$$

Then (i) r^* is the smallest value of r such that

$$C(r) \cap X \neq \emptyset \tag{8}$$

and (ii)

$$x^* \in C(r^*) \cap X. \tag{9}$$

The set $C_i(r)$ is a circle of radius r/w_i centered at A_i. In words, the theorem says that if the circles all start small and are allowed to grow simultaneously then the first point at which they all intersect with the feasible region and with each other is the optimal solution. Furthermore, this is a globally optimal solution regardless of the structure of the feasible region. The set X can be nonconvex or even disconnected.

The theorem would perhaps be of little value in the usual location-analysis setting of linear, nonlinear and combinatorial optimization. However, in the context of interactive computer graphics, it immediately provides a recipe for a globally optimal procedure. The algorithm is exactly as described by the theorem, to let the circles grow (in inverse proportion to their weights) until the optimality condition (8) holds, with determination of whether or not the condition holds made visually by a person at the computer graphics screen.

1.3 The Tektronix 4010/DEC System 10 Implementation. In the pre-AIRMICS phase of this work at The University of Tennessee, the algorithm described above was implemented on a Tektronix 4010 interactive graphics terminal with a DEC System 10 host computer. The Tektronix 4010 has a direct view black and white screen. An electronic tablet and stylus are used in this implementation for data entry and control of the algorithm. A menu of commands [Appendix,

Figure 1] is provided on the tablet so the user can govern the circle expansion in the interactive graphical algorithm.

The Tektronix 4010 is an "unintelligent" terminal which means that each new display must be drawn afresh regardless of any resemblance to the previous display. For this reason and because the 4010 draws all displays as sequences of straight line segments, several program features to hasten convergence of the algorithm were needed. These features, also activated by touching the light pen to the menu, are in summary as follows: (i) Because Helly's theorem [15] guarantees that the unconstrained optimum will be determined by a subset of at most three existing facilities, the user can specify such subsets for trial solutions. In this way, the user's intuition can be exploited to drastically reduce the number of iterations of the algorithm. (ii) Since "circles" on the graphics screen are actually drawn as regular polygons, and since drawing time depends mostly on the number of straight lines in the display, the user can control the number of sides in the polygons. Small values are used for speed in early iterations, large values for accuracy in the latter interations. (iii) The user can delete existing facilities that evidently will not determine the optimum. (iv) The user can zoom in to subregions of the graphics screen when the vicinity of the optimum is recognized.

These features, which require interactive control, just as the overall algorithm does, were pivotol to the success of the implementation. However, the DEC System 10 host computer is a heavily utilized multiprocessor, and, as noted, the 4010 is a rather unsophisticated graphics terminal. Thus a more advanced implementation was deemed essential before serious consideration could be given to Army applications of the system.

2. Enhancements

The major portion of effort that went into the author's STAS was for the development of enhancements to the original Brady/Rosenthal interactive graphical location analysis system. The enhancements are subdivided into three sections in this report as follows:

- The first section describes a reimplementation of the system on much more advanced equipment than was used for the original implementation.
- The second section reports on features of the new implementation of the Brady/Rosenthal algorithm that improve on the original implementation in terms of (i) speed; (ii) accuracy, and (iii) the size of minimax location problem that can be solved. These new features are called programming enhancements.
- The third section reports on algorithmic enhancements, which enable the new implementation to solve new classes of problems that could not be handled previously.

2.1 The Chromatics CG1999 Advanced Implementation. Task (a) of the three specific tasks that comprised the author's STAS was to "study the opportunities for improvements and enhancements that would be enabled by converting from . . . the Textronix 4010 to an advanced-graphics environment." This study was concluded and all of its recommendations were incorporated in the work performed by Mr. Brady on his concurrent LRCP.

The advanced-graphics equipment chosen for the reimplementation of the interactive location analysis system was the Chromatics CG1999 housed at the School of Industrial and Systems Engineering, Georgia Tech. The Chromatics CG1999 has a self-contained Z-80 processor with BASIC interpreter, 50,000

bytes of dynamic random access memory, 3 color planes (hence $2^3=8$ colors), a blink plane, and a 512 by 512 raster with 60 HZ refresh. The auxiliary hardware is a light pen, a digitizer pad and a dual floppy disk drive (although one disk is sufficient for the reimplemented location analysis program). Although Georgia Tech personnel have established a communication link between the Chromatics and a CYBER 74, there is no need for a host computer with the location program.

The fact that the Chromatics implementation of the location algorithm is "stand-alone" is an obvious major advantage over the Tektronix 4010/DEC 10 implementation. Another advance of considerable importance is color. In the author's opinion, color is a worthwhile graphics tool only when simultaneous display of more than one form of information is required. This capability is exploited to maximum potential in Mr. Brady's program. The three color planes have different purposes and can be displayed simultaneously without confusion to the user. The colors and their uses are as follows:

- The red plane contains the menu, from which commands are activated by the light pen.
- The blue plane contains the circles. This is the dynamic portion of the graphics display. It can change independently of the other planes.
- The green plane contains the "background," that is, the map or other pictoral representation that distinguishes feasible and infeasible locations.

The Chromatic's ability to superimpose these three planes is a key element, in terms of human factors, of the superiority of the new implementation over the original one. Having the menu on the screen rather than on a tablet means that the user can see exactly what he or she is controlling when the

commands are activated. This should substantially reduce the likelihood of error. Furthermore, the use of superimposition saves time and fatigue since the user's eyes are not constantly changing focus back and forth between the screen and the tablet. Superimposition of the independent planes is largely responsible in yet another way for the extensive improvement that the Chromatics brought to the interactive graphical location algorithm. A critical deficiency in the Tektronix 4010 implementation was that backgrounds had to be redrawn each time the circles were redrawn. This deficiency was avoided by the use of different color planes for circles and for backgrounds.

Background information may be input to the screen either directly with the light pen in "paint" mode or by retrieval from disk of a file containing a previously entered background. Creation of background files is possible in two ways: (i) a "save" command can be issued from the menu to save the current display on the green plane, or (ii) a map, aerial photograph or other two-dimensional representation can be traced over the digitizer pad. Background files can of course be modified by use of a combination of the two creation methods.

2.2 Programming Enhancements. Documentation of the Chromatics CG1999 implementation of the interactive graphical location analysis program has been prepared by Mr. Brady and submitted to AIRMICS concurrently with this report. A complete list of the programming enhancements achieved through the conversion to an advanced-graphics environment follows. Mr. Brady's documentation should be consulted for greater detail.

- Light-pen control.
- Optional command menus for various portions of the analysis, with optional superimposition of the menus (in red) over the circles (in blue) and backgrounds (in green).

- Optional display of solution parameters.
- Interchangeable scales for numerical input.
- Optional superimposition of backgrounds with fast generation and continuous refreshment.
- Background "save" and "paint" commands on menu with internal disk I/O routines.
- Choice of digitizer pad or light-pen for background entry device.
- Choice of manual or automatic circle expansion/contraction with user specification of rate in automatic mode.
- Facility entry command simplified by use of constant Euclidean/retilinear modes.
- Facility deletion command simplified by giving the user the option of touching either the center or the edge of a circle.
- "Center ID," a new command to identify a critical center with a touch of an edge.
- "Facilities Currently Served," a new command which may be used in conjunction with "Center ID." It reports the level of service that would be given to each existing facility if a new facility were located at the identified point or other user-specified point. Blinking tic marks indicate which existing facilities are adequately served. This feature enables the user to attempt a solution without adhering to the rules of the algorithm.
- "Complex Boundary Fill," a new command to shade arbitrary intersection regions. This feature is used in conjunction with an algorithmic enhancement to be described in the next section.
- Improved program logic for user convenience and error prevention including

- Default actions.
- User prompting with activation of the blink plane in appropriate sections of the menu.
- User reminders: blinking tic marks inside an activated menu command if command execution depends on subsequent light-pen entries.
- Crash proofing: the program ignores spurious light-pen entries; illegal commands prompt error messages and reentry requests.

2.3 Algorithmic Enhancements. Task (b) of the STAS was to "study the expected needs and characteristics of potential Army applications of interactive location analysis . . . and to propose specific improvements and enhancements to meet these needs and characteristics." This study has been completed, and it resulted in the identification of five specific areas of need. The following is a compendium of these needs and the status of research to meet them. In brief, the first two of the needs have been met; the remaining three have not.

- Multifacility problems. A major limitation of the original Brady/Rosenthal algorithm was that all the problems it could solve were allowed to have only one new service facility. This deficiency would be critical in the radio problem, for example, if it was deemed unwise or impractical to attempt to have a single receiver monitoring all the signals. A reasonable scenario might have a set of m ($m>1$) monitors that cover all the signals collectively but not individually. An optimal extension of the Brady/Rosenthal algorithm has been achieved for this situation and its implementation on the Chromatics is reported later in this document.

- Generalized weighting functions. The physical interpretation of the circular set $C_i(r)$ is that it is the region containing all locations from which the new facility can give adequate service for the i^{th} existing facility, within an expenditure of effort as measured by r. In all the location problem formulations considered up to now, it has been possible to represent the cost of service between a new facility and an existing facility as a linear function of the interfacility distance (Sometimes a transformation as between (3) and (5) is needed.) The mathematical consequences of this representation are that the radius of $C_i(r)$ increases linearly with r and the rate of this increase is constant at $1/w_i$. There may be some problems for which the cost of service cannot be measured linearly and for which "regions-of-adequate-service" do not grow so regularly with r. Some analysis of the theorem underlying the Brady/Rosenthal algorithm revealed that constant weighting is not an essential premise. In general, suppose a problem is posed with the cost of service between two points x and A_i represented as $w_i(d(x,A_i))$ where w_i is a function that maps the positive reals into themselves. Then the regions $C_i(r)$ would be defined

$$C_i(r) = \{x : w_i(d(x,A_i)) \leq r\} \tag{10}$$

and all that is necessary for the theorem to hold is for the w_i functions to be increasing. Therefore, the original algorithm can be used to optimally solve the extended problem.

- Network-based modeling. A large portion of the location analysis literature considers problems where the underlying mathematical structure is a network as opposed to the plane. These models can more accurately

represent the distances between facilities, at the cost of considerably more involved data preparation. The extension of the interactive graphics concept to this kind of problem would be very desirable but would require a major algorithmic overhaul. This is relegated to future research by Mr. Brady and the author. Some ideas for how this research should be undertaken and some elaboration on the benefit of network modeling are given in Section 3.2.

- Extensions to non-minimax objectives. Many location problems are more appropriately modeled with miniSum or other objective function forms, rather than minimax. It would be an extremely valuable extension to the interactive location system, therefore, if these non-minimax forms could be accommodated. There is no apparent way to do this without sacrificing optimality. Designing good heuristics for minimum and other non-minimax cases is another important element of future research.
- Data modeling and report generating modules. To be fully operational for the Army, the interactive graphical location analysis system should be equipped with a set of preprocessors (data modeling modules) and postprocessors (report generating modules) that are tailored to specific applications. As high a degree as possible of specificity should be aimed for. No work as yet has been undertaken in this regard because the exact nature of future Army applications is not yet known.

2.4 The Multifacility Algorithm. There are several possible extensions of model (4-5) that involve multiple new facilities. Let m be the number of new facilities and let x_j be the location in R^2 of the j^{th} new facility. The existing-facility locations (called "points" sometimes for simplicity) are still denoted by A_i and the respective weights by w_i. It is assumed that each

point obtains its service from the nearest new facility. Thus, given $\underline{x} = (x_1,\ldots,x_m)$, the cost incurred for serving the i^{th} existing facility is

$$\underset{j=1,\ldots,m}{\text{minimum}}\ w_i\, d(x_j,A_i). \tag{11}$$

The worst served existing facility then has weighted distance

$$\underset{i=1,\ldots,n}{\text{maximum}}\ \ \underset{j=1,\ldots,m}{\text{minimum}}\ w_i d(x_j,A_i) \tag{12}$$

The most direct extension of the minimax philosophy of (4-5) is to choose $\underline{x}$ so as to

$$\text{minimize}\ \ \underset{i=1,\ldots,n}{\text{maximum}}\ \ \underset{j=1,\ldots,m}{\text{minimum}}\ \ w_i d(x_j,A_i) \tag{13}$$

subject to

$$x_j \varepsilon X, \quad j=1,\ldots,m, \tag{14}$$

where, as before, X is the set of feasible locations.

If this problem were posed on a network instead of on the constrained plane, then it would be the well-studied "m-center problem" [12]. In working on this problem, it became clear that a certain weakness exists in the typical m-center model. The weakness carries over to model (13-14) as well. It is the seemingly unrealistic implication within (13) that service levels for the points not providing the maximum in (13) are irrelevant. It became clear that an algorithm for the multifacility problem would not only have to concern itself with providing the best possible service for the "critical," worst-case existing facility, but also it must be concerned with providing good service for the other points.

The strategy taken in this regard is the philosophy of <u>lexicographic optimization</u>. The idea of this approach is the following. If there exist several solutions that have the worst-served point as well-served as possible, then from among those solutions, choose one that serves the second worst-served point as well as possible. If there are several solutions that tie with respect to this secondary criterion, then break the tie according to the tertiary criterion of serving the third worst-served point as well as possible, and so on. Lexicographic optimization as an approach to multiobjective optimization is obviously limited to cases with massive "tying" (which is equivalent to dual degeneracy in a mathematical programming context). The multifacility minimax location problem is evidently such a problem because there are numerous ways to locate the "non-critical facilities" (i.e., facilities not serving the worst-case points) once the critical facility(ies) is (are) located.

The tactics for implementing the lexicographic approach will be given later. First, it is necessary to optimize the primary objective, that is, to minimize the maximum weighted service distance. Again the circles $C_i(r)$ offer the key insight to the problem. Consider a very small problem with m=2, n=4, and suppose r has a current value such that

$$C_1(r) \cap C_2(r) \neq \emptyset \qquad (15)$$

and

$$C_3(r) \cap C_4(r) \neq \emptyset \qquad (16)$$

but all other pairwise intersections are empty. This intersection pattern reveals that if the two new facilities are placed one in each of the nonempty intersection regions, then all four points will be served within weighted

distance r. Obviously, then r should be decreased, so as to improve the worst case. As r is decreased, the two intersections regions (15) and (16) will shrink down to single points. That is, the circle pairs $\{C_1(r), C_2(r)\}$ and $\{C_3(r), C_4(r)\}$ will become tangent. Let $r^{(2)}$ be the value of r when the first of these tangencies occurs. If $r<r^{(2)}$ then one of the conditions (15-16) no longer holds, so that it is no longer possible to have two new facilities serve all four points within weighted distance r. Therefore, $r^{(2)}$ is the optimal objective function value and the point of the first tangency is the optimal location for one of the facilities. This facility is, in fact, the critical facility.

But where should the second (noncritical) facility be placed? Locating it anywhere within the remaining intersection region will yield a solution to the problem. The lexicographic idea for this example is implemented rather simply. The two points that are served by the critical facility (that is, the points whose circles yielded the first tangency) should be deleted and the remaining two circles should be contracted until they are tangent. The point of tangency is evidently the most equitable location for the second facility.

In general, the lexicographic concept is: once the critical facility is found in the m facility problem, delete the points that are served by this facility and then find the critical facility for the m-1 facility problem over the remaining points. Then delete the points served by this facility and solve an m-2 facility problem, etc. Repeat until all m facilities are located.

This algorithm is a recursive application of the original Brady/Rosenthal algorithm. Its success in implementation depends on the user's ability to perceive whether or not a given intersection pattern of circles $C_i(r)$ yields an m-facility coverage of the points. In general, define

$r^{(m)}$ = optimal objective function value for m facility problem

$k(r)$ = smallest number of facilities from which service to all points can be given within weighted distance r.

$k(r)$ will be called the coverage number. Note

$$k(r^{(m)}) = m \tag{17}$$

$$k(r) \leq m \text{ if } r > r^{(m)} \tag{18}$$

$$k(r) > m \text{ if } r < r^{(m)}. \tag{19}$$

The algorithm can start with r=0, so that k(r)=n; that is, each point requires its own facility. (Equivalently, $r^{(n)}=0$.) The user would then command an increase in r and when the first intersection occurs, the value $r^{(n-1)}$ is achieved, that is, k(r)=n-1. Further increase of r would result in more intersections and further decrement of k(r) till eventually k(r)=m. (However, one cannot assume that each new intersection yields one more unit decrease in k(r).)

An equally valid alternative is to start with $r=r^{(1)}$ as determined by the single facility algorithm and to decrease r until $r=r^{(m)}$. The question remains as to how to determine k(r). There are two possibilities. In limited experimentation (with nonrandom subjects), it appeared that users looking at the graphics screen could determine the coverage number visually. The use of the Complex Boundary Fill Feature on page 10 facilitates this process. The user can request shading in any intersection region to keep track of which regions contain facility locations when computing a coverage number. The

second approach, relying less on user capabilities, is to formulate the set covering problem

$$\text{minimize } k(r) = \sum_{\alpha} y_{\alpha}$$

$$\text{s.t. } \sum_{\alpha} a_{i\alpha} y_{\alpha} \geq 1, \; i=1,\ldots,m$$

$$y_{\alpha}=0 \text{ or } 1$$

where α is an index for the set of regions of the screen as partitioned by the circles, excluding those regions that do not intersect with x. The variable $y_{\alpha}=1$, if region α is selected to contain a facility. The coefficient $a_{i\alpha}=1$ if region α is contained in $C_i(r)$;$a_{i\alpha}=0$, otherwise. This is a well-solved problem and should be especially easy to solve because considerable advantage can be gained from application of cover-matrix reduction techniques.

3. Future Work

Task (c) of the author's STAS is to "develop specifications for the . . . enhanced system, and propose the design of a field experiment for demonstration and evaluation."

3.1 Applications. The interactive graphical location analysis system on the Chromatics CG1999 is designed for general use rather than specific applications. As noted in Section 2.3, the development of high-specificity preprocessing and postprocessing modules should accompany any planned applications. The following are some areas of Army operation that are possibly able to employ the interactive location system. These were identified in consultation with USA-CSC personnel. (See Section 5.)

- Logistical support for gun batteries.
- Logistical support for ammunition dumps.

- Missile battery commands.
- Field communications.
- Supply support.
- Control of transportation units.
- Artillery command.
  - Field artillery.
  - Air defense.

3.2 Extensions. As noted under Algorithmic Enhancements, there are three important areas for future extensions: non-minimax objective functions, network-based modeling, and high-specificity pre-/postprocessing modules. The first two extensions require extensive creative effort on the part of operations research scientists. Mr. Brady and the author anticipate that these projects will fit into their overall research program in interactive location analysis as follows:

- Phase I (The University of Tennessee): Development of concept of interactive graphical location analysis; optimal solution of constrained single-facility minimax problems; Tektronix 4010 implementation. Status: completed.
- Phase II (AIRMICS): Chromatics CG1999 advanced implementation; optimal extensions for problems with multiple new facilities and/or generalized weighting functions. Status: completed.
- Phase III (sponsor unknown): Thorough development of network-based modeling capability. Status: conceptual groundwork in progress. (See below.) Concurrent activity: supervision and coordination of efforts to produce application-specific pre/postprocessor modules. Status: identification of application area not yet made.

- Phase IV (sponsor unknown): Incorporation of non-minimax objectives with attempt to extend optimality theorem; probably resort to heuristics; performance measures on heuristics. Continuation of concurrent activity of Phase III. Status: no progress to date.

At this point, the conceptual groundwork for the network-based modeling extension will be given. A great deal more data input effort goes into a network location model than into a planar location model. This issue deserves extensive consideration in Phase III but has not as yet been worked on. The value of network modeling is that the network distance measure can explicitly incorporate very general constraints on the locations of facilities and on the routes to be taken during interfacility travel. For example, if a mine field is situated between a new facility and a point it must serve, then the true travel distance is not the Euclidean or rectilinear distance. Rather, the distance corresponds to whatever is deemed the shortest safe route. Network modeling is the only technique with the capacity to incorporate this important consideration.

The algorithmic complication introduced with network modeling is that the set $C_i(r)$ no longer has regular geometry. In essence, the regular geometry of Euclidean and rectilinear circles is what makes the original interactive graphics algorithm tractable. Our idea for overcoming this difficulty is to develop a rapid procedure for finding all points in the network within distance r of a given vertex, i.e., a "generalized circle." To be efficient, this procedure will have to employ fairly sophisticated data structures for network calculations. Once the generalized circle (which is actually a subnetwork) is identified, its depiction on the screen will be colored or made to blink or be enclosed by some convex closed curve. It is not clear

which of these alternatives will be taken; perhaps some human factors experiments will be done to determine which method of highlighting circles is most condusive to user identification of intersections.

3.3 Experimentation. The interactive location analysis system has a great deal of dependence on user reliability. This is true of all interactive systems, but probably more so than usual in this case because of the system's numerous devices for exploiting user intuition. These devices are included not only because high-level users are anticipated, but also because of the following philosophical position of the author. The design of computer programs for people who have no computer specialization is a burgeoning field. Too much of this work is based on too little respect of the nonspecialist users' intelligence. As long as a convenient interface exists, the user should be expected to do considerable amounts of learning and creative thinking after a very short time.

Given the high expectations on user capability that went into the design of this system, it is essential that thorough testing be made of actual user performance. Some quantification of recorded, controlled user performance would be very valuable. Furthermore, such tests may lead to the discovery of deficiencies in the person/computer interface and result in some redesign of the menu or protocol. The ideal subjects for this experimentation would be logistics officers assigned to the Computer Systems Command. Their performance would be measured in terms of the accuracy of their solutions to randomly generated problems and the time required to reach these solutions. Error rate sensitivity and timing sensitivity to the following factors would be analyzed:

- Form of instruction given in program usage: brief vs. extensive, oral vs. written.

- Problem size, i.e., number of points.
- Weight variance: lower error rates would be expected when the weights are nearly equal.
- Irregularity of feasible region.

Another potentially valuable experiment, from the point of view of general DSS research, would be to deprive some of the officers of instruction in the algorithm. They would be given descriptions of the problem to be solved and of the menu commands, and be asked to find the solution by their own creative processes. It seems likely that several subjects would find solutions and it would be interesting to analyze the methods employed.

4. Demonstration and Publication

On August 23, 1979 a preliminary version of Mr. Brady's program on the Georgia Tech Chromatics CG1999 was demonstrated to the following personnel of the Computer Systems Command Support Group:

- Major Victor Burrell, Chief of Plans and Operations.
- Major Jerry Rawlinson, SAILS Project Officer.
- Captain David Brown, SAAS Project Officer.
- Mr. James Tadlock, SAMS Project Officer.

The reactions were favorable with Maj. Rawlinson and Capt. Brown requesting further discussions. They were convinced that tactical and logistical applications of the system were likely to emerge in the future. Their input was solicited for the identification of applications areas, the results of the query appearing in Section 3.1. Major Rawlinson was also contacted in September at which time he indicated considerable interest on the part of Mr. Graham McBryde, a GS-14 at Fort Lee who is heavily involved in Army applications of computer graphics.

The research paper in the Appendix has been submitted for publication to A.I.I.E. Transactions (A.I.I.E. = American Institute of Industrial Engineers) and is still in the refereeing process. Assuming this paper is accepted, a follow-up paper by Brady, Rosenthal and Dr. Donovan Young of AIRMICS will be submitted for publication. This paper, with acknowledgement to AIRMICS, will report on the Chromatics CG1999 advanced implementation, the multifacility algorithmic enhancement and the generalized weighting functions.

APPENDIX

The main problem addressed in this paper is

$$\underset{x \in X}{\text{minimize}}\ \underset{i}{\text{maximum}}\ w_i d_i(x, A_i) \tag{1}$$

where

x = variable location (vector of coordinates) of a new facility

A_i = known location of i^{th} existing facility, $i = 1, \ldots, n$

w_i = known positive "interaction weight" of i^{th} existing facility

X = known subset of R^2

d_i = Euclidean or rectilinear metric.

In reference to the voluminous literature of location theory (as taxonomized by Francis and White [11]), the problems treated here are single-facility, weighted, constrained, minimax, in the plane, and with Euclidean, rectilinear or mixed metric. "Mixed metric" refers to the case of d_i being not all the same and "constrained" means that the set X of feasible new-facility locations can be a proper subset of R^2. There are no imposed restrictions on the structure of X; for example, X can be a nonconvex region.

The practical importance of minimax location problems has been discussed by numerous authors including those cited in references [1-12, 14, 20, 27]. In brief, minimax formulations are related to "equity" considerations, particularly when the new facility provides emergency service; whereas minisum formulations, which have the maximum in (1) replaced by a summation, relate to "efficiency" considerations.

Allowing the decision maker to specify an arbitrarily shaped feasible region is the primary contribution of this paper. Though it rendered impracticable existing mathematical-programming based facility location algorithms, the importance in practice of such an extension is self-evident. This extension led to the development of an interactive computer

graphical location procedure that is globally optimal under very mild assumptions on human performance. Interactive computer graphical optimization techniques have also been reported for routing [17, 18, 19], scheduling [13], transportation planning [21, 22, 23] and unconstrained search [25], but without assertions of optimality.[1]

Review of Minimax Location Analysis in the Plane

Elzinga and Hearn [7] presented an elegant geometric $O(n^2)$ algorithm for problem (1) in the case of $X=R^2$, $w_i \equiv 1$ and $d_i \equiv$ Euclidean, and they presented a closed-form solution for the case of $X=R^2$, $w_i \equiv 1$ and $d_i \equiv$ rectilinear. (Dearing [3] attributed this relative ease in the rectilinear case to Helly's theorem [15]). Nair and Chandrasekaran [20] also developed an efficient algorithm for the unweighted unconstrained Euclidean case, and later Shamos and Hoey [24] developed an $O(n \log n)$ algorithm using Voronoi diagrams. Dearing [4] has shown that the Voronoi diagram approach can be extended to the weighted unconstrained Euclidean case;

---

[1] An assertion of optimality for an interactive procedure may be controversial to some readers. One might argue that human fallibility makes it unreasonable to ever say "this interactive procedure is optimal if the human-executed steps are performed correctly." The point of view taken here is contradictory to this argument. It is mathematically valid to assert that a sequence of correctly performed steps achieves a certain effect, regardless of the agents of the steps. Such an assertion may lack importance, but not validity, if the agents are unreliable; and it is thus necessary to back the assertion with as strong a guarantee of agent reliability as possible. This issue should not be looked on as being relevant only to interactive algorithms. A computational procedure with a valid optimality proof may fail because of round-off error or some other insufficiency in the computer-executed steps. Devices for controlling numerical error in a linear programming package are analogous to the measures taken to simplify and verify the human's tasks in the present work. In both cases, the probability of error is reduced but not eliminated.

Chandrasekaran and Pacca [1] have also treated this case, with extensions of the approaches in [7] and [20].

Multifacility minimax location algorithms can be specialized to treat cases of problem (1). For example, Chatelon, Hearn and Lowe's [2] subgradient algorithm can solve (1) when $X=R^2$ and the d_i are all the same. Elzinga, Hearn and Randolph's [9] dual nonlinear programming algorithm can solve (1) when $X=R^2$ and $d_i \equiv$ Euclidean. Wesolowsky [27] proposed parametric linear programming for a rectilinear multifacility minimax problem although Elzinga and Hearn [8] pointed out that the parametric analysis could be avoided. Dearing and Francis' [5] multifacility minimax location procedure, based on linear programming duality and network flows, can solve rectilinear cases of (1) with upper bound constraints on the $d_i(x,A_i)$.

All the problems solved by the algorithms cited above have convex objective functions and, in distinction with the present work, convex feasible regions. There has been a great deal more research done on minimax location than is referenced, however most of the other work (e.g., [6, 12, 14]) concerns location on a network (as opposed to in the plane) and has little bearing on the problem at hand. Problem (1) is fundamentally different from the problem obtained from (1) by redefining X as the points of a network and d_i as the shortest path metric. In particular, if the unconstrained problem is defined on a nontree network, then it is not a convex programming problem [6].

Characterization of an Optimal Location

The case of (1) when $X=R^2$, $w_i \equiv 1$ and $d_i \equiv$ Euclidean is commonly called the "minimum covering circle" problem, because the optimal x is the

center of the smallest circle that encloses all the A_i. The radius of the minimum covering circle is the minimum of $f(x) = \max_i w_i d_i(x,A_i)$, in this case. This characterization of optimality does not extend to weighted or constrained problems, so an alternate geometric characterization, due to Francis [10], will be used.

For $r \geq 0$, define

$$C_i(r) = \{x | w_i d_i(x,A_i) \leq r\}, \qquad i=1,\ldots,n, \tag{2}$$

and

$$C(r) = \bigcap_{i=1}^{n} C_i(r). \tag{3}$$

Note that $C(r)=\phi$ iff $f(x)>r$ for all $x \epsilon R^2$. It follows that the unconstrained minimum $\hat{r}$ of $f(x)$ is the smallest r such that $C(r) \neq \phi$. Furthermore, if $\hat{x} \epsilon C(\hat{r})$ then $\hat{x}$ is an unconstrained minimizer. Similarly, the constrained minimum r^* is the smallest r such that

$$C(r) \cap X \neq \phi \tag{4}$$

and an optimal new facility location x^* satisfies

$$x^* \epsilon C(r^*) \cap X. \tag{5}$$

The geometric interpretation of a possible solution procedure is: let the disks $C_i(r)$ have a small (possibly zero) radius initially such that (4) does not hold; then, by increasing r, simultaneously expand the disks until they and X all have a point in common.

An application of the minimum covering circle problem noted by Nair and Chandrasekaran [20] is the siting of a new radio antenna at x to receive signals from transmitters at $A_1, \ldots, A_n$. It was assumed in [20] that the signals originate with equal strength; and the antenna's cost, to be minimized, is a decreasing function of the weakest signal it must

receive. If the originating signal intensities are allowed to differ, taking on values $I_1, \ldots, I_n$, then the problem of maximizing the weakest signal at x is

$$\max_{x \in R^2} \min_i \quad I_i / [d_i(x, A_i)]^2, \tag{6}$$

which is equivalent to (1) with $X=R^2$, $d_i \equiv$ Euclidean and $w_i = 1/\sqrt{I_i}$. In this example $C_i(r)$ is the "listening region" of the transmitter at A_i for an antenna of given receptive strength, and $C(r)$ is the set of feasible locations at which all the transmitters' signals can be received with this antenna. Throughout the "expanding disk" procedure, described above, the radii of the disks $C_i(r)$ are in inverse proportion to the weights, indicating how a strong transmitter has a larger listening region than a weak transmitter. An antenna siting problem is used to illustrate the interactive computer graphical algorithm later in this paper. Such problems in practice are likely to have nonconvex or disconnected feasible regions, as the algorithm can handle. One aspect of the antenna problem that is not treated here is the signal interference caused by uneven terrain; perhaps a three-dimensional graphical model is required to account for this phenomenon.

An Interactive Computer Graphical Algorithm

It is possible to solve the general case of (1) by increasing r until (4) holds but a purely computational algorithm based on the expanding disk procedure may not be practical. For each trial value of r, in order to test for intersection of disks $C_i(r)$, such an algorithm would have to solve or show there is no solution to a system of simultaneous quadratic and/or linear inequalities. The system would have one quadratic inequality for each Euclidean disk and four linear inequalities for each rectilinear disk (diamond).

The techniques developed in this paper are not purely computational. They are expanding disk procedures, but instead of requiring the computer to solve the simultaneous inequalities, they require a human to test for intersection visually at a computer graphics terminal. The attractiveness of this idea is that human minds generally have keener pattern recognition abilities than computers. There are several possible implementations of this person/computer interactive approach. The first to be described is an algorithm that is evidently globally optimal under an assumption of infallibility in the human-executed step of the algorithm. (A weaker assumption will be considered shortly.)

As noted, $C(r) \cap X = \phi$ for all $r < r^*$; and, since $r > r^*$ implies $C_i(r) \supset C_i(r^*)$, $C(r) \cap X \neq \phi$ for all $r \geq r^*$. This monotonicity leads to a bisection search algorithm for r^*:

Step 0. $LB=0$. $UB=M$ where M is some finite number greater than r^*.

Step 1. $r = 1/2\ (UB-LB)$.

Step 2. If $UB-LB$ is within a prescribed tolerance, stop.

Step 3. Display X and $C_i(r)$, $i=1,\ldots,n$ graphically and have the human determine if $C(r) \cap X \neq \phi$.

Step 4. If $C(r) \cap X \neq \phi$, set $UB=r$; otherwise set $LB=r$. Go to Step 1.

Provided an initial UB can be found and assuming correct determinations in Step 3, this interactive algorithm is a globally convergent procedure for finding r^*. (See Garfinkel, Neebe and Rao [12] for a network facility location algorithm that also uses binary search to find the smallest attainable value of a minimax objective function.) An easily computed initial UB is the diameter of the graphics screen divided by the smallest w_i.

Before dealing with the human infallibility assumption, let the algorithm be refined slightly, as follows. Let there be two sequential executions of the algorithm, Phases I and II, in the solution of any instance of the location problem. Phase I identifies an unconstrained optimum; Phase II, a constrained optimum. Phase I starts with LB=0, UB=M; Phase II starts with UB=M and LB equal to the final r value of Phase I. The two phases require different tasks of the user in Step 3: in Phase I it is to determine if the n sets $C_i(r)$ intersect; in Phase II it is to determine if only two sets, X and C(r), intersect.

It is argued that the infallibility assumption is entirely supportable in Phase II. This is because irrespective of n the set C(r) is very easily recognized: the curve-and line-segments enclosing the $C_i(r)$ partition the graphics screen, and C(r) is the (convex) parition that contains the unconstrained optimal location(s). (E.g., see Figure 4.) Infallibility may also be supportable in Phase I (especially after the enhancements to the basic graphical system are introduced), but a preferred tack to this argument may be to insert an efficient verification step for catching human errors in Phase I Step 3. If nonintersection is indicated, the verification step requires the user to identify a nonintersecting disk pair; otherwise, the user is required to identify a point in C(r) (preferably with a light pen). With this additional information, it is simple for the computer to automatically confirm or deny the user's determination.[2]

[2] A person could, of course, deliberately confound the verification procedure. It is assumed that the user will base his intersection/nonintersection indication on a perceived nonintersecting disk pair or a perceived point in C(r), and will report consistently with this perception in the verification step.

Enhancements to the Interactive Graphical System

The assumptions on human performance that guarantee optimality of the algorithm of the last section are: (i) consistent responses in the verification step in Phase I and (ii) correct indications in the simple recognition step in Phase II. These are rather mild, if not lax, assumptions; persons using the interactive graphics system should far surpass these minimum requirements. In fact, the next issue addressed is how to exploit the user's additional capabilities in the graphics system design and thereby achieve faster solutions. In addition, some system features are presented to reduce the probability of human error.

When speaking of "solution times" in reference to an interactive program, elapsed time is usually far more relevant than CPU time. For example with the program described here, even though measures were taken to reduce elapsed time at the expense of CPU time, the ratio of the former to the latter was in the range 50-150. In comparison, the ratio of CPU cost vs. user wage is probably in the range 5-30 and decreasing.

The best techniques for reducing elapsed time in an implementation of the interactive location algorithm will depend on the graphics equipment used in that instance, but a number of generally applicable ideas can be presented. The most important point to stress is that the person operating the interactive program can be expected to very quickly develop an intuitive ability to predict the optimal locations. Furthermore, this intuition, which is almost always observed in interactive graphics applications [16, 28], can be exploited without sacrificing the global optimality of the interactive location algorithm. The required modification of the algorithm is to give the user the option of overriding the computer in Step 1, subject to computer verification of $LB<r<UB$ for the user-selected r. This modification is especially helpful in Phase I.

The Phase I problem, to determine the unconstrained optimum $\hat{x}$, can be shown by Helly's theorem [3, 26] to satisfy the following property: there exists a subset $S \subset N = \{1,\ldots,n\}$ with $2 \leq ||S|| \leq 3$, such that

$$\min_{x \in R^2} \max_{i \in S} w_i d_i(x,A_i) = \min_{x \in R^2} \max_{i \in N} w_i d_i(x,A_i) \tag{7}$$

and $w_i d_i(\hat{x},A_i)$ is constant over S. For any doubleton or tripleton S, the left-hand-side minimization in (7) can be computed accurately in small constant time without human interaction. After a few expanding-disk iterations, the user is likely to have some good guesses of the correct set S for equality in (7). He can input this set to a program feature that automatically sets r to the left-hand-side of (7) and tests (7) for equality. If equality holds, $\hat{x}$ is reported. Otherwise, the program continues with the new r value, which is closer to $\hat{r}$ than the old, and the user can form a better guess of S by observing which facilities caused the violation of (7).

This technique for reducing the number of iterations will save time in any implementation of the interactive location algorithm. The degree of improvement obtained from the other time-saving program enhancements to be reported is hardware-dependent. These other features are aimed at reducing the amount of time the user has to spend waiting for graphic displays to be drawn. Drawing time is the major component of elapsed time, especially when using an "unintelligent" graphics terminal which requires each display to be drawn afresh regardless of any resemblence to the previous display. Three ways to save drawing time follow: (i) Since "circles" on the graphics screen are actually drawn as regular polygons, and since drawing time depends mostly on the number of straight lines in

the display, let the user control the number of sides in the polygon. Small values would be requested for speed in the early iterations, large values for accuracy in the latter iterations. (ii) At intermediate iterations the user is likely to recognize some of the existing facilities that will be irrelevant in determining the optimum, e.g., a facility whose disk subsumes another's. Let the user request deletion of such facilities from subsequent displays (but continue to include all facilities in the verification step). (iii) Let the user "zoom in" on a subregion of the graphics screen when the vicinity of the optimum is recognized. The drawing of Figures 7 and 8, illustrating this feature, took 50 and 5 seconds, respectively. These features that save drawing time also improve human accuracy by giving the user a less cluttered picture to analyze.

Contours, Addends and Layouts

The interactive computer graphics program can be used for solving constrained problems even if the user does not desire to input a graphically defined feasible region. The alternate approach, which has been described for other single-facility location problems by Francis and White [11], is to generate contours of the objective function $f(x) = \max_i w_i d_i(x, A_i)$. Contours enable immediate comparison of alternate proposed sites and are useful for incorporating subjective factors into the location analysis. For all $r > \hat{r}$, the boundary of $C(r)$ is a contour of value r. For example, all points on the highlighted curve of Figure 4 have equal objective function value; points inside the curve have better values; points outside, worse. A program command that results in printout of graphic display is especially useful for contour generation.

In some situations there may not exist clearly defined feasible and infeasible regions. Portions of the plane may have relatively differing

attractiveness as potential sites for new facilities. If these differences can be depicted graphically (with a color terminal, ideally), then contours can be used to help select a location with a most desirable combination of objective function value and site attractiveness.

A problem related to (1) is

$$\min_{x \in X} \max_{i} [d_i(x,A_i) + k_i] \tag{8}$$

where k_i is a nonnegative number called "addend." This model might be appropriate, for example, if x is the location of a new ambulance depot, A_i is a patient location, and k_i is the distance from A_i to the nearest hospital. Goldman [14] suggested that for locating some emergency facilities, models with addends k_i are preferred to models with weights w_i. Goldman's argument against weights is as follows. Suppose the existing facilities are communities and the new facility will give centralized emergency service at a cost shared by the communities. The weight w_i is a measure of the frequency of emergencies at A_i requiring aid from the central facility. Under model (1) a community which expends its own resources to achieve a small weight will be penalized, perhaps unfairly, in terms of proximity to the new facility.

The interactive graphics algorithm can be used as described to solve

$$\min_{x \in X} \max_{i} [w_i d_i(x,A_i) + k_i] \tag{9}$$

be redefining the expanding disks as

$$C_i(r) = \{x \mid d_i(x,A_i) \leq e_i(r)\} \tag{10}$$

where $e_i(r) = \max\,[0, (r-k_i)/w_i]$. (Elzinga and Hearn [7] solved (8) for the cases of $X=R^2$ and $d_i \equiv$ Euclidean or $d_i \equiv$ rectilinear by extension of their techniques for problems without addends).

Facility layout is another area where interactive computer graphics may be effective for problem solving. One of several possible ideas along these lines follows. Of all compact sets $S \subset R^2$ with given area the one that achieves

$$\min_{S} \max_{i} \max_{s \in S} w_i d_i(s, A_i) \qquad (11)$$

is called by Francis [10] a minimax facility design. Francis [10, p. 1166-7] points out that a minimax facility design is the set enclosed by a contour of $f(x) = \max_i w_i d_i(x, A_i)$, such as in Figure 4. This suggests the following computer graphical procedure for finding the minimax design: generate a contour with the graphics program, then test for the desired interior area (either numerically or with a planimeter), then increase or decrease r accordingly (either bisectionally or interactively), and repeat. The set obtained after a small number of iterations will be sufficiently close to the desired area, inasmuch as minimax facility designs are idealized layouts, which in practice might guide in choosing the location and orientation of rectangular layouts.

Example

The interactive location algorithm has been implemented at the University of Tennessee with a Tektronix 4010 graphics terminal on line to a DEC-10 computer. A graphics tablet and electronic stylus are used for program input and control. Commands are issued by touching the stylus to one of the items on the "menu" section of the tablet. The menu illustrated in Figure 1 is sufficient for executing all the program enhancements mentioned in the previous sections. The ZOOM/PAN and HARDCOPY commands would be unnecessary with graphics equipment that has these features built into the hardware. The feasible region and other background information can be input by laying a map, blueprint or other drawing over

the tablet and tracing it out with the stylus.

Figures 2-9 illustrate the solution of hypothetical location problems with the interactive graphics program. Except for the highlighting of the contour and minimax facility design in Figure 4, the figures are computer generated. The figures are photographs of the Tektronix 4010 screen, printed in reverse so that black in the figures represents white on the screen. Figures 2-5 show the solution of Phase I of an antenna siting problem with unequal signal strengths. Figures 6-8 show the Phase II solution, where the feasible region is the perimeter of a house. (The elapsed time for this example was 5 minutes, CPU time was 5 seconds.[3]) Figure 9 shows the final iteration in enlarged scale of a mixed metric problem where the feasible region is "not in the lake."

Conclusion

This paper has presented an interactive graphics program that efficiently locates global optima for some nonconvex problems. The core of the program is a well-defined algorithm with a simple, human-executed pattern recognition task as one of its steps. Because the users of interactive graphical programs are known to develop strong intuition, the program is equipped with interactive options that exploit the intuition and enhance performance. Location analysts and others who apply mathematical programming are likely to find additional instances in the future when interactive graphics lead to global solutions of previously intractable problems.[4]

[3]These times are not presented as representative. Differences in graphics hardware will be most significant in accounting for time variation; other factors will be user familiarity, problem characteristics and load on the time-shared computer.

[4]Helpful comments on an earlier version of this paper were received from Richard Francis, Robert Garfinkel, Jean-Paul Jacob, John A. White and Donovan Young.

References

1. Chandrasekaran, R. and M.J.A.P. Pacca, "Weighted Min-Max and Max-Min Location Problems: Finite and Polynominally Bounded Algorithms," University of Texas at Dallas, 1977.

2. Chatelon, J., D.W. Hearn, and T.J. Lowe, "A Subgradient Algorithm for Certain Minimax and Minisum Problems," Mathematical Programming 15, 130-145 (1978).

3. Dearing, P.M., "Minimax Location and Helly's Theorem," Guest Seminar, Management Science Program, University of Tennessee, Knoxville, 1977.

4. Dearing, P.M., "Minimax Location and Voronoi Diagrams," ORSA/TIMS Atlanta Meeting, 1977.

5. Dearing, P.M. and R.L. Francis, "A Network Flow Solution to the Multifacility Minimax Location Problem Involving Rectilinear Distances," Transportation Science 8, 126-141 (1974).

6. Dearing, P.M., R.L. Francis and T.J. Lowe, "Convex Location Problems on Tree Networks," Operations Research 24, 628-642 (1976).

7. Elzinga, J. and D.W. Hearn, "Geometric Solutions for Some Minimax Location Problems," Transportation Science 6, 379-394 (1972).

8. Elzinga, J. and D.W. Hearn, "A Note on a Minimax Location Problem," Transportation Science 7, 100 (1973).

9. Elzinga, J., D.W. Hearn and W.D. Randolph, "Minimax Multifacility Location with Euclidean Distance," Transportation Science 10, 321-336 (1976).

10. Francis, R.L., "Some Aspects of a Minimax Location Problem," Operations Research 15, 1163-1169 (1967).

11. Francis, R.L. and J.A. White, Facility Layout and Location: An Analytic Approach, Prentice-Hall (1974).

12. Garfinkel, R.S., A.W. Neebe, and M.R. Rao, "The m-Center Problem: Minimax Facility Location," Management Science 10, 1133-1142 (1977).

13. Godin, V.B., "Interactive Scheduling: Historical Survey and State of the Art," AIIE Transactions 10, 331-337 (1978).

14. Goldman, A.J., "Minimax Location of a Facility in a Network," Transportation Science 6, 407-418 (1972).

15. Hadwiger, H., H. Debrunner and V. Klee, Combinatorial Geometry in the Plane, Holt, Rinehart and Winston (1964).

16. Jacob, J.-P., "Potential of Graphics to Enhance Decision Analysis," Proceedings Second Lawrence Symposium on Systems and Decision Sciences, October 3-4, 1978, p. 292-300.

17. Krolak, P., W. Felts, and G. Marble, "A Man-Machine Approach Toward Solving the Traveling Salesman Problem," Communications of the A.C.M. 14, 327-334 (1971).

18. Krolak, P., W. Felts, and J. Nelson, "A Man-Machine Approach Toward Solving the Generalized Truck Dispatching Problem," Transportation Science 6, 149-170 (1972).

19. Krolak, P., and M.R. Williams, "Computerized School Bus Routing," Tech. Report 78-1, Computer Science, Vanderbilt University, (1978).

20. Nair, D.P.K. and R. Chandrasekaran, "Optimal Location of a Single Service Center of Certain Types," Naval Research Logistics Quarterly 18, 4 (1971).

21. Rapp, M.H., "The Planning of Node Oriented Transit Systems: An Application of Man-Machine Interactive Problem Solving," Socio-Economic Planning Sciences 6, 95-123 (1972).

22. Schneider, J.B., D.G. Miller and T. W. Friedman, "Locating and Sizing Park-Ride Lots with Interactive Computer Graphs," Transportation 5, 389-406 (1976).

23. Schneider, J.B., J.G. Symons, Jr. and M. Goldman, "Planning Transportation Terminal Systems in Urban Regions: A Man-Machine Interactive Problem-Solving Approach," Transportation Research 6, 257-273 (1972).

24. Shamos, M.I. and D. Hoey, "Closest-Point Problems," Proceedings of 16th Symposium on Foundations of Computer Science, IEEE, 151-162, October 1975.

25. Tsuda, T., "Interactive Search for the Global Maximum of a Function of Many Variables," IEEE Transactions on Automatic Control AC-22, 397-403 (1977).

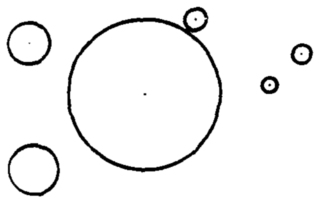
26. Walker, R., "The Captains' Meeting: Finding an Optimal Rendezvous for n Ships," Mathematical Gazette 62, 263-267 (1978).

27. Wesolowsky, G.O., "Rectangular Distance Location Under the Minimax Optimality Criterion," Transportation Science 6, 103-113 (1972).

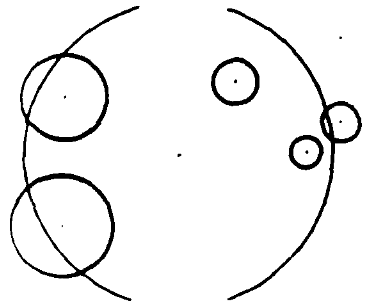
28. Young, D., "Interactive Optimization," Guest Seminar, Computer Science and Management Science Departments, University of Tennessee, Knoxville 1979.

DELETE EXISTING FACILITY	ENTER BACKGROUND	ZOOM/PAN	HARDCOPY	SELECT TWO-FACILITY SUBSET	STOP
ENTER EXISTING FACILITY	RECTILINEAR (DEFAUT = EUCLIDEAN)	ORIGINAL SCALE	CHANGE POLYGON PARAMETER	SELECT THREE-FACILITY SUBSET	START/ CONTINUE

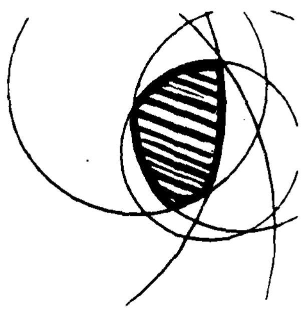
Figure 1. Menu of commands for control of interactive graphics program



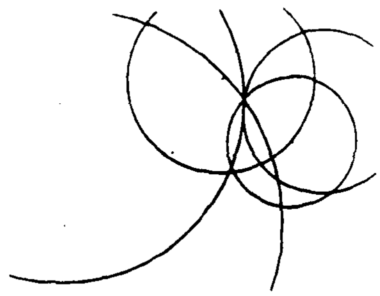
2



3

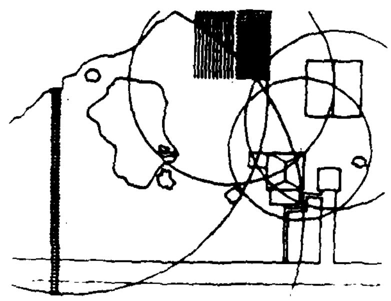


4

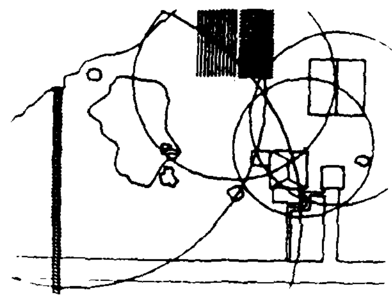


5

Figures 2 through 5. Phase I of the interactive graphical algorithm: solving an unconstrained problem. An objective function contour is highlighted and a minimax facility design, C(r), is shaded in Figure 4.

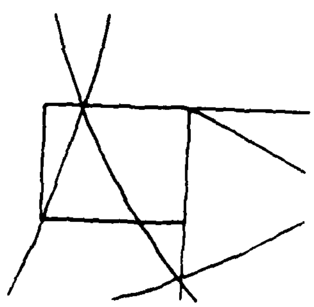


6



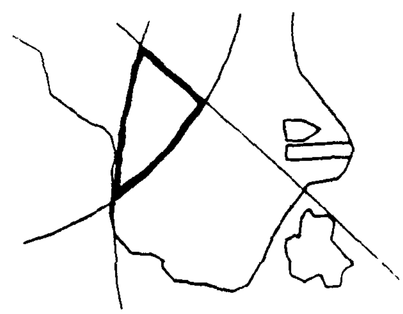
7

Figures 6 and 7. Phase II of interactive graphical algorithm: solving a constrained problem.



8

Figure 8. Close-up of constrained optimum of Figure 7



9

Figure 9. Constrained optimum for mixed metric problem.

END

DATE

FILMED

10-81

DTIC